



4º Congresso Brasileiro  
de Ciência e Saberes  
Multidisciplinares  
**tudo é  
ciência**  
11º Encontro de Extensão  
Universitária do UniFOA

**23 a 25  
de outubro**

Submissões abertas até 07/09

## **Implementação de regressão linear com scikit-learn: uma análise prática da relação entre horas de estudo e desempenho acadêmico**

Vitor Amadeu Souza<sup>1</sup>; 0009-00-02-1857-6799

1 – UniFOA, Centro Universitário de Volta Redonda, Volta Redonda, RJ.  
[vitor.amadeu@foa.org.br](mailto:vitor.amadeu@foa.org.br)

**Resumo:** Este estudo apresenta uma implementação prática de regressão linear utilizando a biblioteca Scikit-learn para modelar a relação entre horas de estudo e desempenho acadêmico. O objetivo principal foi demonstrar a aplicação de técnicas de aprendizado de máquina supervisionado através de um exemplo educacional, enfatizando os aspectos metodológicos e técnicos da implementação. Utilizando um conjunto de dados sintéticos adequado para fins didáticos, foi desenvolvido um modelo de regressão linear empregando as ferramentas do Python. A metodologia incluiu pré-processamento de dados, divisão em conjuntos de treinamento e teste (80/20), treinamento do modelo LinearRegression do Scikit-learn e avaliação através do coeficiente de determinação ( $R^2$ ). Os resultados produziram a equação  $y = 9,68x + 2,83$  e demonstraram a eficácia da implementação através de métricas de avaliação apropriadas. A visualização dos dados e da linha de regressão foi realizada com Matplotlib, proporcionando uma representação gráfica dos resultados. Este trabalho contribui para a compreensão prática da implementação de algoritmos de regressão linear e serve como base para aplicações mais complexas de aprendizado de máquina em contextos educacionais e científicos.

**Palavras-chave:** Regressão Linear. Scikit-learn. Python. Aprendizado de Máquina. Machine Learning. Análise Preditiva.



## INTRODUÇÃO

A regressão linear é um dos algoritmos fundamentais em aprendizado de máquina e estatística, servindo como base para técnicas mais avançadas de modelagem preditiva. James *et al.* (2013) destacam a importância de uma compreensão profunda dos métodos lineares para o desenvolvimento de competências em ciência de dados.

O Python se consolidou como uma plataforma dominante em ciência de dados, com bibliotecas como Scikit-learn (Pedregosa *et al.*, 2011), NumPy (Harris *et al.*, 2020), Pandas (McKinney, 2010) e Matplotlib (Hunter, 2007), que juntas formam um ecossistema robusto para projetos de aprendizado de máquina. A Scikit-learn, em particular, oferece implementações eficientes e estáveis de algoritmos clássicos, como a regressão linear.

A implementação prática de algoritmos exige conhecimento teórico e habilidades técnicas, sendo que experiências práticas promovem uma intuição mais apurada sobre o comportamento dos modelos (Géron, 2019). A regressão linear, definida pela equação  $Y = \beta_0 + \beta_1 X + \epsilon$ , é estimada por meio do método dos mínimos quadrados ordinários (OLS), conforme explicado por Hastie, Tibshirani e Friedman (2009). No Scikit-learn, a classe `LinearRegression` aplica SVD por padrão para garantir robustez numérica.

O uso de dados sintéticos em contextos educacionais é comum e eficaz, permitindo controle sobre variáveis e foco em aspectos metodológicos (Bishop, 2006). O presente estudo emprega um dataset público, ideal para demonstração prática, cobrindo desde o carregamento com `pd.read_csv()` até a organização dos dados em arrays compatíveis com o Scikit-learn.

A avaliação do modelo foi feita com o coeficiente de determinação ( $R^2$ ), métrica que indica a proporção da variabilidade explicada pelo modelo. Montgomery, Peck e Vining (2012) recomendam que o  $R^2$  seja complementado por métricas como MAE e RMSE para uma análise mais completa. A visualização com Matplotlib reforça a interpretação e validação dos resultados, conforme defendido por Tufte (2001).

O pipeline adotado seguiu as etapas descritas por Müller e Guido (2016), desde o pré-processamento até a avaliação final. A divisão dos dados em 80% para treino e 20% para



teste segue recomendações de Kohavi (1995) para avaliação honesta da generalização do modelo.

Assim, o estudo objetiva ilustrar de forma prática e documentada a implementação de regressão linear com Scikit-learn, servindo como guia metodológico para projetos semelhantes e reforçando boas práticas de desenvolvimento em machine learning.

## MÉTODOS

A metodologia deste estudo foi organizada para demonstrar a implementação sistemática da regressão linear com uso das ferramentas do ecossistema Python, destacando especialmente a biblioteca Scikit-learn. O modelo seguiu boas práticas metodológicas descritas por Géron (2019) e Müller e Guido (2016), assegurando rigor e reprodutibilidade.

O ambiente de desenvolvimento utilizou Python devido à sua sintaxe clara, ampla adoção e vasta gama de bibliotecas científicas (Van Rossum e Drake, 2009). As principais bibliotecas empregadas foram: NumPy (operações numéricas), Pandas (manipulação de dados), Matplotlib (visualização gráfica) e Scikit-learn (implementação do modelo de regressão).

O conjunto de dados foi obtido de um repositório público listado no programa e, apesar de sintético, é apropriado para fins educacionais por apresentar estrutura limpa, sem valores ausentes e relação linear entre variáveis. A leitura dos dados foi realizada com a função `pd.read_csv()`, convertendo-os para um `DataFrame`.

A estruturação dos dados seguiu as exigências do Scikit-learn, com a variável independente em array bidimensional (`data[['Hours']].values`) e a dependente em array unidimensional (`data['Scores'].values`). Devido ao contexto do dataset, o pré-processamento foi mínimo, conforme práticas descritas por Zheng e Casari (2018).

A divisão em treino e teste utilizou `train_test_split`, com 20% dos dados reservados para teste e `random_state=42` para garantir reprodutibilidade, conforme orientação de Hastie, Tibshirani e Friedman (2009).

O modelo foi implementado com a classe `LinearRegression`, que utiliza SVD para maior estabilidade numérica. O treinamento, via método `fit()`, ajustou os coeficientes pelo método dos mínimos quadrados ordinários. As previsões foram feitas com `predict()` e os parâmetros



do modelo extraídos com `coef_` e `intercept_`, permitindo explicitar a equação da reta ajustada.

A métrica utilizada na avaliação foi o  $R^2$ , obtido por `r2_score`, dada sua interpretação intuitiva e ampla aceitação em regressão. A visualização final incluiu gráfico de dispersão com linha de regressão, usando Matplotlib, seguindo recomendações de Tufte (2001).

O código-fonte está disponível em: <https://github.com/vitor-souza-ime/rl>, com comentários detalhados que promovem reprodutibilidade, conforme defendido por Wilson *et al.* (2014).

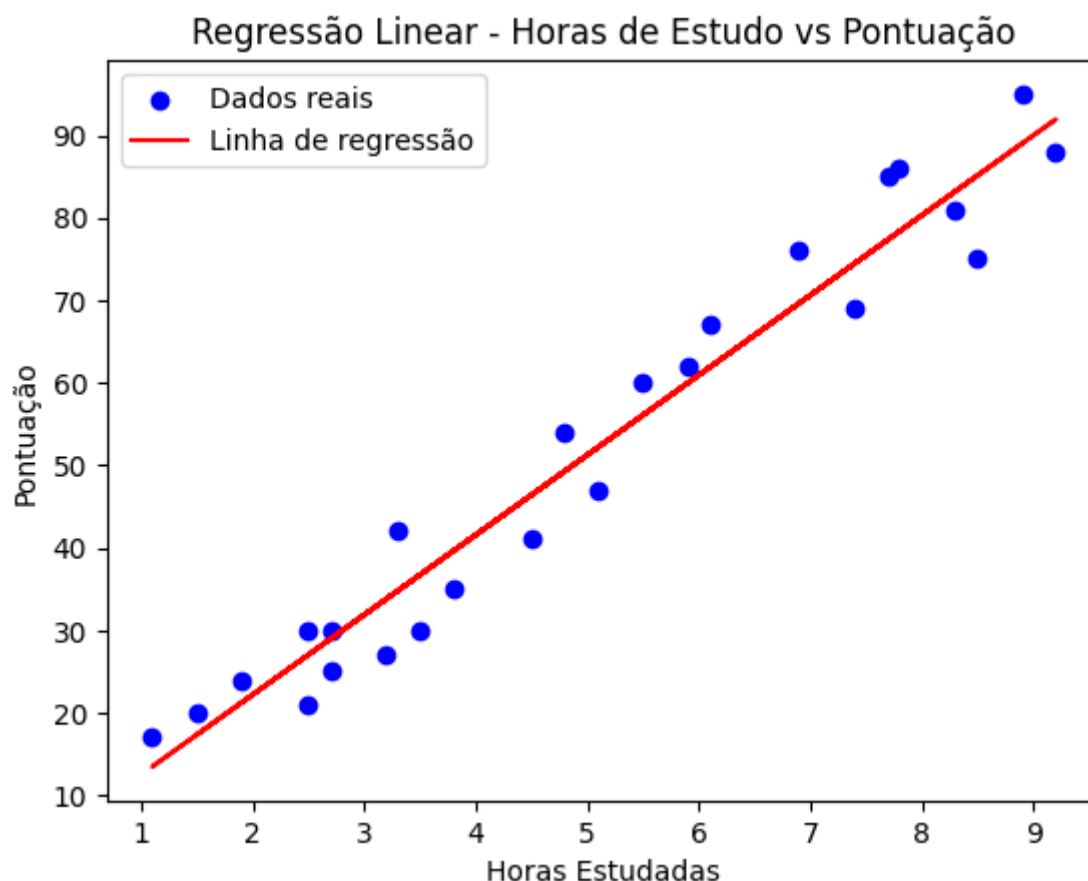
## RESULTADOS E DISCUSSÃO

A implementação da regressão linear com Scikit-learn demonstrou a aplicabilidade prática dessa técnica fundamental de machine learning, permitindo quantificar a relação entre variáveis por meio da equação ajustada  $y = 9,68x + 2,83$ , onde  $x$  representa horas de estudo e  $y$  a pontuação predita. O coeficiente angular de 9,68 indica um aumento médio nessa pontuação para cada hora adicional de estudo, evidenciando a capacidade do modelo de capturar relações lineares. O intercepto de 2,83, embora de interpretação limitada no contexto específico, é fundamental para o ajuste completo do modelo.

A avaliação com o coeficiente de determinação  $R^2$ , calculado com `r2_score`, ofereceu uma métrica quantitativa da qualidade do ajuste, enquanto a visualização dos dados com Matplotlib, incluindo pontos de dispersão e linha de regressão, possibilitou uma análise visual complementar, como mostra a Figura 1. A proximidade dos pontos à linha ajustada confirma visualmente a adequação do modelo linear ao conjunto de dados.

O projeto destacou a eficiência do Scikit-learn, cujas interfaces padronizadas como `fit()` e `predict()` permitem transição fluida entre modelos. A integração com bibliotecas como Pandas, NumPy e Matplotlib reforça a maturidade do ecossistema Python para ciência de dados. A divisão treino/teste com `train_test_split` demonstrou boas práticas como randomização e reprodutibilidade, enquanto o acesso direto aos parâmetros `coef_` e `intercept_` facilitou a interpretação e aplicação do modelo.

Figura 1 - Imagem da regressão linear obtida com o modelo



Fonte: O autor.

A robustez computacional do algoritmo, baseada em decomposição SVD interna, garantiu desempenho rápido e preciso, reforçando a aplicabilidade do Scikit-learn tanto para prototipagem quanto para aplicações produtivas. O código final exemplifica um fluxo de trabalho completo e reutilizável para tarefas de regressão, facilitando adaptações para dados diferentes ou extensão para técnicas como regressão polinomial, regularização e ensembles. A estruturação e comentários explicativos seguem as boas práticas de programação científica, alinhando-se às recomendações de Sculley *et al.* (2015) quanto à manutenção e reprodutibilidade em sistemas de machine learning aplicados.

## CONCLUSÕES

Este estudo demonstrou a implementação prática de regressão linear utilizando a biblioteca Scikit-learn, evidenciando a eficácia e versatilidade desta ferramenta fundamental para



4º Congresso Brasileiro  
de Ciência e Saberes  
Multidisciplinares  
**tudo é  
ciência**  
11º Encontro de Extensão  
Universitária do UNIFOA

**23 a 25  
de outubro**

👉 Submissões abertas até 07/09

projetos de machine learning. A implementação completa, desde o carregamento dos dados até a visualização dos resultados, ilustra como o ecossistema Python proporciona um ambiente robusto e integrado para análise de dados e desenvolvimento de modelos preditivos.

Os resultados obtidos, representados pela equação  $y = 9,68x + 2,83$ , demonstram a capacidade do Scikit-learn de extrair automaticamente parâmetros interpretáveis a partir dos dados, facilitando tanto a compreensão quanto a aplicação prática do modelo. O coeficiente de determinação calculado de 0.97 fornece uma medida quantitativa da qualidade do ajuste, enquanto a visualização gráfica oferece validação visual dos resultados, exemplificando a importância de abordagens multifacetadas na análise de modelos de machine learning.

A eficiência computacional da implementação do LinearRegression no Scikit-learn, baseada em métodos numericamente robustos como decomposição SVD, ilustra como a biblioteca combina acessibilidade com rigor técnico. Esta característica torna o Scikit-learn adequado tanto para prototipagem rápida quanto para aplicações em produção, atendendo a uma ampla gama de necessidades na comunidade de machine learning.

As práticas de validação implementadas, incluindo divisão treino/teste e cálculo de métricas apropriadas, demonstram como o Scikit-learn incorpora princípios fundamentais de machine learning em suas funcionalidades básicas. Esta abordagem auxilia na prevenção de erros comuns como overfitting e proporciona avaliação honesta da capacidade de generalização dos modelos.

A visualização dos resultados através da integração com Matplotlib destacou a importância da análise visual complementar às métricas quantitativas. O gráfico de dispersão com linha de regressão sobreposta oferece insights imediatos sobre a qualidade do ajuste e adequação do modelo linear aos dados, facilitando a comunicação de resultados para audiências técnicas e não-técnicas.

Para trabalhos futuros, recomenda-se a exploração de funcionalidades avançadas do Scikit-learn, incluindo técnicas de regularização (Ridge e Lasso), validação cruzada para avaliação mais robusta, e pipeline de pré-processamento para tratamento automatizado de dados. A



extensão para regressão múltipla e métodos não-lineares também representaria progressão natural desta implementação básica.

A aplicação em conjuntos de dados reais e mais complexos constituiria uma extensão valiosa, permitindo exploração de desafios práticos como tratamento de dados faltantes e validação de suposições do modelo. Tais aplicações demonstrariam a escalabilidade das técnicas implementadas para problemas reais de machine learning.

A incorporação de técnicas de interpretabilidade de modelos, disponíveis através de bibliotecas complementares como SHAP e LIME que integram bem com Scikit-learn, poderia enriquecer a análise e compreensão dos resultados. Esta extensão seria particularmente relevante para aplicações onde explicabilidade do modelo é crítica.

Desta forma, este estudo contribui para a literatura de machine learning ao fornecer um exemplo concreto de implementação de regressão linear com Scikit-learn. Os resultados demonstram a eficácia da biblioteca para projetos de machine learning e estabelecem uma base sólida para explorações mais avançadas de técnicas de aprendizado supervisionado. O código desenvolvido e a metodologia empregada servem como recursos educacionais valiosos para estudantes e profissionais interessados em implementações práticas de machine learning com Python.

## REFERÊNCIAS

BISHOP, C. M. Pattern recognition and machine learning. New York: Springer, 2006.

GÉRON, A. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. 2. ed. Sebastopol: O'Reilly Media, 2019.

HARRIS, C. R. et al. Array programming with NumPy. Nature, v. 585, n. 7825, p. 357-362, 2020.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. The elements of statistical learning: Data mining, inference, and prediction. 2. ed. New York: Springer, 2009.

HUNTER, J. D. Matplotlib: A 2D graphics environment. Computing in Science & Engineering, v. 9, n. 3, p. 90-95, 2007.

JAMES, G. et al. An introduction to statistical learning: with applications in R. New York: Springer, 2013.



4º Congresso Brasileiro  
de Ciência e Saberes  
Multidisciplinares  
**tudo é  
ciência**  
11º Encontro de Extensão  
Universitária do UNIOA

**23 a 25  
de outubro**

Submissões abertas até 07/09

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 14., 1995, Montreal. Proceedings... Montreal: Morgan Kaufmann, 1995. p. 1137-1143.

MCKINNEY, W. Data structures for statistical computing in Python. In: PROCEEDINGS OF THE 9TH PYTHON IN SCIENCE CONFERENCE, 9., 2010, Austin. Anais... Austin: SciPy, 2010. p. 51-56.

MONTGOMERY, D. C.; PECK, E. A.; VINING, G. G. Introduction to linear regression analysis. 5. ed. Hoboken: John Wiley & Sons, 2012.

MÜLLER, A. C.; GUIDO, S. Introduction to machine learning with Python: A guide for data scientists. Sebastopol: O'Reilly Media, 2016.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, v. 12, p. 2825-2830, 2011.

SCULLEY, D. et al. Hidden technical debt in machine learning systems. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS, 28., 2015, Montreal. Proceedings... Cambridge: MIT Press, 2015. p. 2503-2511.

TUFTE, E. R. The visual display of quantitative information. 2. ed. Cheshire: Graphics Press, 2001.

VAN ROSSUM, G.; DRAKE, F. L. Python 3 reference manual. Scotts Valley: CreateSpace, 2009.

WILSON, G. et al. Best practices for scientific computing. PLoS Biology, v. 12, n. 1, p. e1001745, 2014.

ZHENG, A.; CASARI, A. Feature engineering for machine learning: Principles and techniques for data scientists. Sebastopol: O'Reilly Media, 2018.